

Glenn Procedural Requirements

GLPR 7150.1A

Effective Date: June 27, 2011

Expiration Date: June 27, 2016

COMPLIANCE IS MANDATORY

This Document Is Uncontrolled When Printed.

Validate prior to use at <https://knowledgeshare.grc.nasa.gov/bmslibrary>

**Responsible Office: Code D/ Engineering and Technical
Services Directorate**

GRC Software Engineering Requirements w/Change 1 (4/11/2012)

TABLE OF CONTENTS

Change History	2
Preface	3
Chapter 1. Introduction	6
Chapter 2. Responsibilities	8
Chapter 3. Project and Software Life-cycles	10
Chapter 4. Project Life-cycle	12
Chapter 5. Process Improvement	22
Appendix A. Definitions	23
Appendix B. Acronyms	26
Appendix C. Requirements Mapping Matrix	27
Appendix D. Standard Process Tailoring Guidelines	41
Appendix E. Standard Guidance	42

Distribution: BMS Library

Change History

Change	Date	Description/Comments
Basic	06/29/07	Document converted from CLP (GRC-P2.6.4) to GLPR; updated to encompass all of NPR 7150.2. Added chapter 7 to cover commercial/government/modified off-the-shelf (COTS/GOTS/ MOTS). Added chapter 8 to cover acquisition. Revised entire document to replace Control Level references with 7150 Software Classification.
A	06/27/2011	Document updated to NPR7150.2A. Document content rewritten and layout redone.
Change 1	04/11/2012	Changed responsible office from D/ Engineering and Technical Services Directorate to D/Engineering Directorate and added distribution statement on page 1.

Preface

P.1 Purpose

- a. The purpose of this document is to clearly articulate and establish the requirements on the implementing organization for performing, supporting, and evaluating software engineering activities in accordance with NASA Procedural Requirement (NPR) 7150.2, “NASA Software Engineering Requirements.”
- b. All projects creating or acquiring software (code) by or for NASA are required to comply with NPR 7150.2. The requirements in NPR 7150.2 apply to software development, maintenance, operations, retirement, management, acquisition, and assurance activities started after its initial date of issuance. If a project or program intends to apply practices that do not meet the requirements in NPR 7150.2, then the project is required to apply for a waiver with the appropriate Software Technical Authority. Any approved waivers are required to be documented. However, *all* waivers should be documented; refer to the NASA John H. Glenn Research Center (GRC) Technical Authority Implementation Plan, GRC_TAIP_010610, available from the BMS Library, for information on GRC’s plan for implementing technical authority. Refer to Software@Glenn for the names of the software technical authorities at GRC.

P.2 Applicability

- a. This Glenn Procedural Requirement (GLPR) applies to all NASA GRC Space, Aeronautics, and Research programs, projects, subprojects, and tasks that are developing or acquiring software. This includes infrastructure related (facility control software, ODIN support software, etc.) software (code) efforts. This GLPR also applies to the development of complex electronics up to the test procedures if the complex electronics has a processor in it.
- b. A project in the context of this GLPR is defined as any effort that includes software developed, acquired, or maintained by NASA GRC. These efforts may range from NPR 7120.5 defined programs to sub-projects managed through other NASA Centers.
- c. This GLPR is the procedure for software providers and software acquirers. Contracting the development of software is covered in chapters 4. This GLPR also applies to software reuse and inclusion of Off-The-Shelf (OTS) software. Use of Commercial Off-The-Shelf (COTS), Government Off-The-Shelf (GOTS) and Modified Off-The-Shelf (MOTS) software is covered in chapters 4 and 5 of this document. If incorporating previously developed software into software with a higher classification, follow GRC-SW-7150.10, “Transition of Software to a Higher Classification.” If the software will be released outside of GRC, follow GRC-SW-7150.11, “Preparing Software for Release.” This GLPR applies when a software simulation or model is to be developed at GRC as a project, subproject, or task. The procedure for using existing simulation software, however, is documented in GLPR 8080.2, “Implementation - Analytical Simulation” and is not covered here.
- d. This GLPR references processes and templates that have been written to assist the project in achieving compliance with NPR 7150.2, but they are not mandatory for all projects (see appendix E for the GRC-SW-7150.xx series of standard processes and templates, along with a brief description of each). Projects that choose not to use these processes are still required to meet the requirements from NPR 7150.2 that pertain to the classification of their software. A mapping between GRC processes and procedures and NPR 7150.2 requirements is provided on the Software@Glenn Web site (<http://software.grc.nasa.gov>, “GRC Processes to 7150 Requirements.xls”).

- e. This GLPR and GLPR 8739.1, “Software Assurance,” are the basis for compliance with the software assurance requirements area of NPR 7150.2.
- f. The requirements of this document are applicable to existing programs and projects started after the initial date of this GLPR.
- g. This procedure is a combination of GRC requirements and project implementation best practices. GRC requirements are identified with “**shall**” statements,” a good practice by “should,” permission by “may” or “can,” and descriptive material by “is,” “are,” or other forms of the verb “to be.” All other material is considered guidance that can be applied as needed to the project.

P.3 Authority

- a. NASA-STD-8719.13, “NASA Software Safety Standard”
- b. NASA-STD-8739.8, “NASA Software Assurance Standard”
- c. NPD 7120.4, “NASA Engineering and Program/Project Management Policy”
- d. NPR 7120.5, “NASA Space Flight Program and Project Management Requirements”
- e. NPR 7120.7, “NASA Information Technology and Institutional Infrastructure Program and Project Management Requirements”
- f. NPR 7120.8, “NASA Research and Technology Program and Project Management Requirements”
- g. NPR 7123.1, “NASA Systems Engineering Processes and Requirements”
- h. NPR 7150.2, “NASA Software Engineering Requirements”
- i. NPR 8000.4, “Agency Risk Management Procedural Requirements”

P.4 Applicable Documents

- a. GLPR 1050.2, “Internal Customer Agreements”
- b. GLPR 1270.1, “Corrective and Preventive Action”
- c. GLPR 1440.1, “Records Management”
- d. GLPR 7120.5.20, “GRC Project Deviation/Waiver Process”
- e. GLPR 7120.0.6, “Lessons Learned Capture Process”
- f. GLPR 7123.22, “Product Verification and/or Validation”
- g. GLPR 8080.2, “Implementation - Analytical Simulation”
- h. GLPR 8730.5, “Glenn Research Center Business Management System Quality System Manual”
- i. GLPR 8739.1, “Software Assurance”

P.5 Measurement/Verification

- a. The Software Technical Authority conducts assessments of programs/projects to verify compliance with this document at each major milestone review. Compliance will be determined by reviewing the archived artifacts required by this procedure.
- b. Independent internal and external audits of this procedure are also performed as part of the overall GRC Business Management System (BMS) Quality System process per GLPR 8730.5, “Glenn Research Center Business Management System (BMS) Quality System Manual.”

- c. Projects are required to classify their software into Classes A, B, C, D, E, F, G, or H in accordance with the software classification definitions of appendix E in NPR 7150.2. Projects developing software in classes A, B, C, D, F (not OTS), and G (not OTS) are required to collect metrics and produce metrics reports per NPR 7150.2. Several resources are available on the Software@Glenn Web site to assist in complying with the NPR 7150.2 metrics requirements. These resources are listed in appendix E herein, along with brief descriptions of their content.
- d. Each project with software components is required by NPR 7150.2, requirement Software Engineering (SWE)-125, to maintain a compliance matrix tracing the project processes and artifacts to requirements in NPR 7150.2. In order to assist projects in establishing and maintaining this traceability, a template for each software class has been created and can be found on the Software@Glenn Web site. This template contains the mapping of NPR 7150.2 requirements to this procedure and its referenced processes. It includes a column for listing the location of project artifacts that show compliance.
- e. Programs/projects are to provide comments regarding the use of this procedure. Programs/projects should provide comments/feedback to the Chief Engineer Office (CEO) by use of the Corrective and Preventative Action Reporting System (CPARS) in accordance with GLPR 1270.1 for future updates. The Software Project Lead (SPL) is to provide this feedback to the Software Engineering Process Group (SEPG) if the analysis indicates changes and/or improvements to its standard processes are warranted. This can be done by using the Feedback Form (located at Software@Glenn) or by meeting with the SEPG Lead.
- f. The SEPG will periodically collect project metrics, remove information that identifies those metrics with a particular project, summarize the data, and present the summary data to GRC management, including the GRC Chief Engineer.
- g. GRC is required to periodically perform a Software Inventory for the Headquarters Office of the Chief Engineer (by NPR 7150.2, SWE-006). The intent of the software inventory is to support compliance with NASA policy and facilitate application of resources to mitigate risk. The software inventory contains a list of programs and projects containing software that are under development or in maintenance.

P.6 Cancellation

- a. GLPR 7150.1, GRC Software Engineering Requirements, dated June 29, 2007.

/S/

James M. Free
Deputy Director

Chapter 1. Introduction

1.1 Procedure Introduction

1.1.1 Rationale and Benefit

a. This document describes the requirements from NPR 7150.2 that are applicable to the activities of the Glenn Research Center (GRC) to enable the safe, effective, and correct development of software for use in NASA projects. This document, along with suggested process methods and associated tools and templates, provides an overview of the requirements for software projects to use as a starting point in executing the project's software life-cycle.

1.1.2 Procedure Overview

a. Software projects **shall** comply with all applicable requirements of NPR 7150.2 as determined by their software classifications. All applicable NPR 7150.2 requirements are located in appendix C.

b. This procedure begins with a discussion of responsibilities in chapter 2. Discussion of the NASA project life-cycle and the typical software life-cycle are presented in chapter 3. The major formulation and implementation periods are subdivided into the typical life-cycle phases. This document does not repeat all of the NPR 7150.2 requirements; rather it provides a summary of the key, overarching requirements of NPR 7150.2 by life-cycle phase in chapters 4 and 5. The intent of chapters 4 and 5 is to provide the necessary linkage among the indicated NPR 7150.2 software engineering requirements (SWE), the GRC required activities, and the SEPG-approved standard processes and templates. These processes and templates are designed for use by GRC software engineering projects. They contain useful cross-references to the NPR requirements and Capability Maturity Model Integration (CMMI) best practices.

c. Projects are expected to prepare appropriate defined process documents from these later resources using the tailoring guidelines contained in appendix D.

1.2 Records

This paragraph provides the minimum records required in relation to the classification. Any additional documentation and/or records identified as optional that may be considered as a record are to be negotiated between the program, project, subproject, or task manager, the software project lead, and in some cases, software assurance, and will be documented in the Software Management Plan.

1.2.1 Records Management

a. All projects **shall** produce and maintain the following records:

(1) Project Work/Work Authorization, Requirements Documentation, Safety-criticality Assessment, and Release Authorization.

b. Design Review Records **shall** be produced and maintained for Class A through E projects, and for Class F and G projects that are developed in-house.

c. Review or Board Records **shall** be produced and maintained for Classes A, B, C, D, safety-critical Class E, and Class F. Nonsafety-critical Class E and Class G should evaluate the need for review or board records.

- d. Problem Reports/Corrective Action Reports **shall** be produced and maintained for Class A, B, C, safety-critical D, safety-critical E, F, and H. Nonsafety-critical D, nonsafety-critical E, and G should evaluate the need to develop and maintain problem reports/corrective action reports.
- e. Verification and validation results **shall** be produced and maintained for all classes except nonsafety-critical D and non-safety-critical E.
- f. Design Change Documentation **shall** be produced and maintained for all classes except nonsafety-critical D, nonsafety-critical E, and H.
- g. Exceptions to any records required in items a. through f. **shall** be approved by the Software Technical Authority.
- h. Each project **shall** establish project records and products based on the software classification and maintain these records in a repository under configuration management.
- i. Record retention activities will follow the GLPR 1440.1, “Records Management.”

1.3 Process Effectiveness Metrics

- a. Metrics will be gathered for the project to indicate the effectiveness of this procedure and provide direction to which processes should be improved.
- b. The Software Project Lead (SPL) will assemble the metrics and submit them to the Software Engineering Process Group (SEPG) by sending an email to grc-software-lead@lists.nasa.gov.
- c. The SEPG will use the metrics to identify areas of improvement.
- d. Requirements:
 - (1) Percentage of fulfilled requirements defined in this document; approved, deviated, or waived requirements equal fulfilled requirements (reference appendix C - Compliance Matrix, for the list of requirements).
 - (2) Number of software requirements.
- e. Work Products and Templates:
 - (1) Percentage of related, provided templates used during this process.
 - (2) Number of deliverable/work products created without using provided templates.
 - (3) Number of SEPG processes used during this process.
- f. Configuration Management:
 - (1) Percentage of deliverable/work products submitted to the Configuration Management System.

1.4 Lessons Learned

- a. The Project Manager will collect lessons learned while performing this procedure and provide them to the Safety and Mission Assurance Directorate (SMAD) and the SEPG, which is responsible for this procedural document.
- b. Lessons learned will be handled in accordance with GLPR 7120.0.6, “Lessons Learned.”

Chapter 2. Responsibilities

2.1 GRC Program/Project/Subproject/Task Managers

- a. All organizations that are acquiring, developing, or managing software are responsible for that software and for ensuring its quality and adherence to NASA and GRC requirements and standards. The GRC program/project/subproject/task managers that acquire software through other GRC organizations are responsible for developing an internal customer agreement with that organization by using GLPR 1050.2, “Internal Customer Agreements.”
- b. In addition, program/project managers are responsible for ensuring sufficient resources are secured for software assurance that are to be negotiated with the GRC Safety and Mission Assurance Directorate.

2.2 Software Project Lead

- a. The software project lead (SPL) and the appropriate branch chief are responsible for negotiating project resources with the GRC program/project/subproject/task manager(s). The SPL is responsible for determining the initial classification of the proposed software, managing and delivering the software portion of the project, coordinating project requirements with the program/project/subproject/task manager, reporting status to the branch chief and GRC program/project/subproject/task manager, and reporting metrics following any organizational and GRC metrics processes. In the case where software is the only product for the project, the GRC program/project/subproject/task manager and the SPL may be the same person.

2.3 Software Team

- a. The software team is responsible for carrying out the software project plans to design, build, and test the software. The software team is responsible for following the software management plan and other processes as directed by the SPL. The software team members are responsible for reporting progress to the SPL as defined in the software management plan.

2.4 Assurance Team

- a. The assurance team is responsible for performing an independent classification assessment of the software to verify the SPL’s classification determination, developing and implementing the project’s software assurance plan, reviewing and concurring on the project software safety plans (when required), assessing the software safety criticality, and auditing the project’s performance (including processes and work products). The assurance team includes members from the SMAD, and/or others assigned to execute software assurance activities from within the project.

2.5 Software Technical Authority

- a. The Software Technical Authority (TA) is responsible for providing the guidance to the programs and projects for achieving compliance with NPR 7150.2. The programs and projects are responsible for meeting and complying with the requirements of NPR 7150.2. The TA is responsible for exercising technical authority by setting and controlling technical requirements and approving any deviations from such requirements at the level commensurate with his/her authority. The TA is responsible for approving the software classification as determined by the SPL.

2.6 Chief Engineer Office

a. The GRC CEO includes the overall function of project chief engineers as a separate entity from the project management staff. The Chief Engineer is responsible for the development and implementation of GRC engineering processes and procedures. The Chief Engineer is responsible for auditing the project compliance with the GRC processes and procedures.

Chapter 3. Project and Software Life-cycles

3.1 NASA Project Life-cycle

a. The Phases of the NASA project life-cycle are shown in figure 3.1. The formulation and implementation periods are divided into seven distinct phases (Pre-Phase A thru Phase F). Each phase is typically marked by a Key Decision Point (KDP), which usually is associated with a prescribed major design review. The NPR 7120.5 provides additional detail about the NASA project life-cycle, associated reviews, and KDP's, concerning the formulation and execution of space flight programs and projects. The NPR 7120.7 provides details about the formulation and execution of information technology and institutional infrastructure programs and projects. The NPR 7120.8 provides details about the formulation and execution of research and technology.

NASA Life Cycle Phases	Approval for Formulation		Approval for Implementation		IMPLEMENTATION		
Project Life Cycle Phases	Pre-Phase A: Concept Studies	Phase A: Concept & Technology Development	Phase B: Preliminary Design & Technology Completion	Phase C: Final Design & Fabrication	Phase D: System Assembly, Int & Test, Launch & Checkout	Phase E: Operations & Sustainment	Phase F: Closeout

Figure 3.1-Phases of the NASA Project Life-cycle

3.2 Software Life-cycle

a. A generic life-cycle model for software development is shown in figure 3.2. Once customer needs and concepts are defined, a software development folder is initiated, requirements development and management begins in the formulation part of the software project's life-cycle and extends into the late implementation phases for final builds. Software design begins after the software requirements review, includes an intermediate software preliminary design review, and concludes with a software design baseline at the software critical design review. Initial and final coding, unit and integration testing, and software documentation, acceptance, and delivery dominate the implementation period. Activities that occur throughout the entire life cycle are configuration management, risk management, project monitoring and control, software assurance, and metrics. Typical work products resulting from the design phase include the Software Management Plan (SMP), the Software Requirements Specification (SRS), and the Requirements Traceability Matrix (RTM). The project should specify key decision points when selecting the type of approved software life-cycle (see paragraph 3.3) to be used on the project.

Formulation		Implementation		
				MAINTENANCE
Configuration Management, Risk Management, Project Monitoring and Control, Software Assurance, Metrics				

Figure 3.2-Phases of the Software Life-cycle

b. Refer to chapter 4 for more detailed explanations of expected activities and applicable NPR requirements in each phase of the approved software life-cycle.

3.3 Approved Software Life-cycles

a. The project ***shall*** select and document the life-cycle(s) to be used in GRC software development activities.

b. The following software development life-cycle models are approved for use.

(1) The standard waterfall life-cycle is the primary choice for software development at GRC.

(2) The evolutionary life-cycle is an approved alternate for software development.

(3) The spiral life-cycle is an approved alternate for software development.

(4) The agile life-cycle is an approved alternate for software development.

c. Refer to Software@Glenn for links to more information on the life-cycle models listed above.

3.4 Tailoring of Life-cycles

a. The variety of projects developing software at GRC may benefit by tailoring the basic elements of one of the approved life-cycles, the key decision points and review events, and the major work products in each phase. A given project should determine its appropriate software life-cycle, along with any modifications, according to the needs of the project. The project may apply to the appropriate software technical authority to gain approval for a tailored approach. It is expected that the primary benefits and work products/results will be either achieved as stated, or traded out for other products that are shown to meet the intent of the original.

Chapter 4. Project Life-cycles

4.1 Formulation

a. The formulation phase in the software life-cycle includes the management and planning activities typically necessary for the successful initiation of a project. During this phase typical items like customer needs, system level requirements, make versus buy strategies, overall project and software management plans, a Work Breakdown Structure (WBS), software safety assessments, and primary project deliverables and work products are defined. Use of the approved standard process document, GRC-SW-7150.3, “Software Project Planning” (see appendix E), will ensure the fulfillment of the applicable SWE requirements.

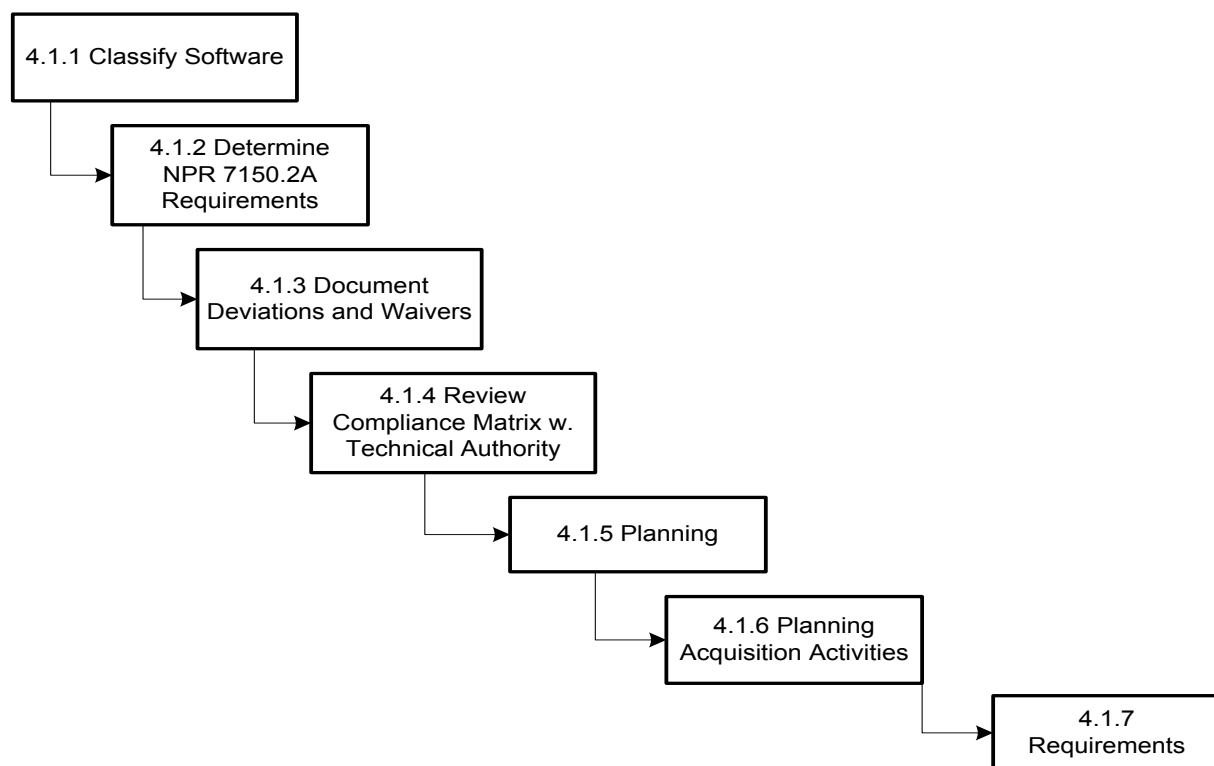


Figure 4.1- Software Project Formulation

4.1.1 Classify and Assess Software

a. The project **shall** classify its software based upon the NPR 7150.2 Classes A thru H.

(1) The Software Technical Authority is responsible for approving software classifications, verifying compliance with NPR 7150.2 and for approval or denial of waivers.

b. This classification **shall** be independently verified by SMAD.

(1) Any disagreement will be resolved with the concurrence of the appropriate Technical Authority and SMAD.

c. The project and SMAD **shall** jointly determine the software safety criticality in accordance with NASA-STD-8739.8, “Software Assurance Standard.”

(1) If the proposed software is determined to be safety critical, refer to NASA-STD-8719.13, “Software Safety Standard” to ensure implementation of applicable software safety requirements, assessment and planning.

(2) Any disagreement will be resolved with the concurrence of the Engineering Technical Authority and SMAD.

4.1.2 Determine Applicability of NPR 7150.2 Requirements

a. The project shall review the Requirements Mapping Matrix (RMM) in appendix C of this GLPR to plan for the activities required based upon the software’s class and safety criticality.

(1) The RMM presented in appendix C contains the same information as the RMM from appendix D of the NPR 7150.2, but groups the requirements by life-cycle phase and process areas so that the project can concentrate on the requirements for that particular phase it is in.

(2) Be aware that the requirements shown in the RMM are the minimal set required for that class. A project should consider other requirements that make sense for the project.

b. During the planning phase, the project shall fill out the NPR 7150.2 Compliance Matrix to indicate how or where it intends to meet each NPR 7150.2 requirement.

(1) A compliance matrix for each software class and safety criticality is available from the Software@Glenn Web site on the controlling documents and authority page.

(2) The RMM tells the project *what* requirements need to be met. The project uses the compliance matrix to indicate *how* it meets those requirements.

c. Projects developing Class A or B software shall comply with all of the requirements listed in the RMM.

(1) The RMM omits Class A and B to save space, since all of the requirements must be met.

d. Projects developing Class C, D, E, F, G, or H software shall comply with the requirements marked by an “X” or “P (Center).”

(1) Be sure to refer to the correct column in the RMM, based upon whether the software is safety critical or not.

e. The project shall work with the Software Technical Authority to determine and document how the project will meet the P (Center) requirements listed in the RMM.

(1) The P (Center) requirements are a nonnull subset of the full requirements. The intent is to allow for the tailoring of the requirements to the unique characteristics of each project.

f. As the project progresses, the project shall update the compliance matrix to provide evidence of how or where it met each NPR 7150.2 requirement.

4.1.3 Document Deviations, Waivers and Tailoring

a. Any deviation or waiver from any SWE **shall** be documented, and subject to the approval of the appropriate (Center or Headquarters OCE) TA.

b. The project **shall** tailor the requirements based upon technical and managerial judgment to provide the necessary level of rigor for the scope and size of the software project.

(1) Projects should follow GLPR 7120.5.20, “GRC Project Deviation/Waiver Process” for deviating or waiving any of the requirements contained in this document requiring approval authority.

c. The reasoning for any tailoring of the requirements **shall** be clearly documented in the compliance matrix.

4.1.4 Review Compliance Matrix with Technical Authority

a. The Software Technical Authority **shall** document its acceptance and approval of the documented Compliance Matrix, including any waivers, deviations, and P (Center) decisions, by signing off on the Compliance Matrix. Approval may also be in the form of an e-mail.

b. Implementation of P (Center) requirements **shall** be fully documented. This documentation will contribute to developing a Center-wide approach, to provide a repository for other projects to consult, and to provide a resource for other technical authorities to consult.

4.1.5 Planning

a. The project **shall** develop plans to use in managing the software development effort. The plans may be separate documents or combined depending on project size and requirements. See the requirements as specified in the RMM for the software class in appendix C.1 and C.2.

b. The Software Management Plan **shall** include at least one cost estimate, and include a schedule of activities with deliverables, milestones, and reviews with completion criteria.

c. A Software Safety Plan **shall** be developed for those projects that include safety-critical software.

d. An Independent Verification and Validation (IV&V) Project Execution Plan **shall** be developed for projects selected by the NASA Chief, SMAD, for IV&V.

e. A Software Configuration Management Plan **shall** be developed.

f. A Software Maintenance Plan **shall** be developed.

g. A Software Assurance Plan **shall** be developed.

(1) For planning and implementation of Software Assurance, refer to GLPR 8739.1, “Software Assurance.”

h. The project **shall** implement and maintain the software plans.

i. Appendix E highlights the following approved standard process documents and templates to aid in project planning:

GRC-SW-7150.3, “Software Project Planning”

GRC-SW-7150.9, “Software Configuration Management”

GRC-SW-7150.13, “Process and Product Quality Assurance”

GRC-SW-7150.16, “Software Estimating”

GRC-SW-TPLT-SAP, “Software Assurance Plan Template”

GRC-SW-TPLT-SMP, “Software Management Plan Template”

GRC-SW-TPLT-SMntP, “Software Maintenance Plan Template”

GRC-SW-TPLT-SCMP, “Software Configuration Management Plan Template”

GRC-SW-TPLT-SSP, “Software Safety Plan Template”

4.1.6 Plan Acquisition Activities

a. The project **shall** fulfill the Acquisition (including OTS) planning requirements as specified in the RMM for the software class in appendix C.3.

(1) Appendix E highlights the following approved standard process documents to aid in acquisition planning:

(a) GRC-SW-7150.14 “Software Acquisition Statement of Work Guidelines”

(b) GRC-SW-7150.15 “Software Acquisition Planning”

4.1.7 Requirements

a. The following are the primary expected practices of this phase of the life-cycle:

Start the requirements phase by transforming system (or higher-level) requirements into software (or product component) requirements. Analyze and document requirements, perform any agreed upon inspections and/or reviews, write the requirements specification that incorporates the safety, system and any specific software development processes into the software requirements. Also, for safety critical software, perform safety analyses on software functional requirements, start software acceptance test planning, begin planning required software operations, maintenance, and retirement activities, document maintenance and retirement activities in the Software Maintenance Plan, and verify/document bidirectional traceability between the software requirements and the higher level requirements in the requirements traceability matrix.

b. The project **shall** document and maintain the requirements in compliance with the Software Requirements Specification template as described in NPR 7150.2 for Classifications A, B, C, and D.

c. The project **shall** fulfill the requirements development activities as specified in the RMM for the software class in appendix C.4.

(1) Appendix E highlights the following approved standard process document and template to aid in requirements development:

(a) GRC-SW-7150.5, “Requirements Development”

(b) GRC-SW-TPLT-SRS, “Software Requirements Specification Template”

(2) If existing software is proposed for reuse in a system or subsystem of higher classification, refer to the approved standard process document, GRC-SW-7150.10, “Transition of Software to a Higher Classification” in appendix E for assistance in requirements development.

d. The project **shall** fulfill the requirements management activities as specified in the RMM for the software class in appendix C.4.

(1) Appendix E highlights the following approved standard process documents and templates to aid in requirements management:

(a) GRC-SW-7150.6, “Requirements Management”

(b) GRC-SW-7150.12, “Formal Inspection and Peer Reviews”

(c) GRC-SW-TPLT-RTM, “Requirements Traceability Matrix Template”

4.2 Implementation

a. The implementation phase in the software life-cycle includes the design, coding, testing and operations activities typically necessary for the successful execution of a project. During this phase typical activities include designing and writing code, validation and verification, component and system assembly and integration, unit and system testing, delivery of final products, and operations.

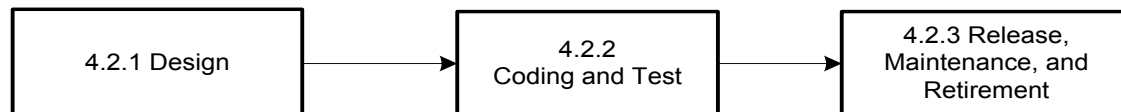


Figure 4-2 Software Project Implementation

4.2.1 Design

a. The following are the primary expected practices of this phase of the life-cycle:

Transform the allocated and derived system and software-level requirements into an architectural design. Document the architectural design(s) using both a static view (module view, component and connector view) and a dynamic view (data flow diagram, finite state machine). For systems with higher classifications, alternative architectures should be developed, documented, and evaluated against the operational scenarios. Perform any rapid prototyping and trade-off studies, select the optimal solution, and capture the results and rationale for selection in the design documentation. Further refine the architectural design into a detailed design depicting lower-level entities that permit implementation by coding in a programming language. Document the detailed design including information such as identifier, type, purpose, function, constraints, subordinates, dependencies, interface, resources, processing, and data. Write software system-level test plans and procedures. Perform agreed upon inspections and reviews. Verify and document bidirectional traceability between the software requirements and the higher level requirements in the requirements traceability matrix.

b. The project **shall** fulfill the design requirements as specified in the RMM for the software class in appendix C.5.

c. The project **shall** fulfill the documentation requirements as specified in the RMM for the software class in appendix C.5.

d. Appendix E highlights the following approved standard process templates to aid in project design:

(1) GRC-SW-TPLT-IDD, “Interface Design Description Template”

(2) GRC-SW-TPLT-SDD, “Software Design Description Template”

4.2.2 Coding and Test

a. The following are the primary expected practices of this phase of the life-cycle:

Implement the design by coding the software. Devise and perform software unit-level and software integration testing in accordance with the software test plans. Document any defects found during testing and track them to closure. Perform any agreed-upon inspections or walkthroughs. Finalize and release integration and software system-level test procedures. Perform verification and validation of software models, simulations, and analysis tools to be used to verify and validate the software. Verify and document traceability between the software design and the resulting code in the requirements traceability matrix. Perform software system-level testing (systems integration, end-to-end, etc.) in accordance with the software test plans. Document any defects found during testing and track them to closure. Document changes to the design, to the requirements, and to the code resulting from the defect closure activities. Document the evaluation of test results in the software test report. For safety critical software, demonstrate, record, and report the safety functions and features. Verify and document traceability between the software requirements and the software test procedures in the requirements traceability matrix.

b. The project **shall** fulfill the code and test requirements as specified in the RMM for the software class in appendix C.6.

c. The project **shall** fulfill the testing requirements as specified in the RMM for the software class in appendix C.6.

d. The project **shall** fulfill the documentation requirements (SWE-114 and SWE-118) as specified in the RMM for the software class in appendix C.6.

e. Appendix E highlights the following approved standard process templates to aid in project coding and test:

(1) GRC-SW-TPLT-STP, “Software Test Plan Template”

(2) GRC-SW-TPLT-STPr, “Software Test Procedure Template”

(3) GRC-SW-TPLT-STR, “Software Test Report Template”

4.2.3 Release, Maintenance and Retirement

a. The following are the primary expected practices of this phase of the life-cycle:

Perform any agreed-upon customer demonstrations/tests in accordance with the software test plans. Document any defects found during testing or demonstrations, and track them to closure. Record and store the version of the release; finalize, review, and sign off on any user documents; get customer agreement on delivery via a release authorization. Provide a version description document and other documentation (i.e., requirements and design documents, user manuals, etc.) needed to support operations and maintenance, as negotiated with the customer during project formulation. Carry out plans for software operations and implement the activities specified in the software maintenance plan such as maintenance and retirement. When planning for software retirement, perform analysis to assess impact to users, stakeholders, and reuse opportunities. Generate a list of what needs to be retired, including software code, documentation, and records, along with proper methods of disposition. Document what format the artifacts are expected to be in (hardcopy or electronic, if electronic, what file format). Establish where the artifacts are going to be stored. Determine who is

responsible for doing the work. Determine what resources will be needed and a mechanism to ensure the resources are available when retirement is reached. Make sure whoever is responsible has the proper training to do the job. Establish a mechanism for informing users of what software applications are being retired. Plan to record any lessons learned from the software.

b. The project ***shall*** fulfill the release, maintenance and retirement requirements as specified in the RMM for the software class in appendix C.7.

c. Appendix E highlights the following approved standard process templates to aid in project release, maintenance, and retirement:

(1) GRC-SW-TPLT-SMntP, “Software Maintenance Plan Template”

(2) GRC-SW-TPLT-SUM, “Software Users Manual Template”

(3) GRC-SW-TPLT-SVD, “Software Version Description Template”

4.3 Supporting Life-cycle Activities

- a. Many activities performed during the project life-cycle span multiple phases. These include risk management, configuration management, software verification and validation, quality assurance, requirements management and traceability, project management, formal inspections and peer reviews, and metrics collection. Thus, during the initial project planning phase, the project conducts planning for the activities, methods, environments, and criteria associated with these activities. These plans are documented in the Software Management Plan (SMP). They are implemented throughout the entire software life-cycle.
- b. For verification and validation, the project records, addresses, and tracks to closure the results of software verification and validation activities. Software test plans and procedures are updated as needed to remain consistent with requirements. Formal inspections and peer reviews are planned and performed for the software products (i.e., requirements, plans, design, and code) produced during each phase of the life-cycle.

4.3.1 Configuration Management

- a. The purpose of configuration management (CM) is to establish and maintain the integrity of work products. Software CM includes the process of applying configuration control throughout the software life-cycle to ensure the completeness and correctness of software configuration items. The CM is to be performed periodically throughout the software project life-cycle.
 - b. The project **shall** fulfill the CM requirements as specified in the RMM for the software class in appendix C.10.
- (1) Appendix E highlights the following approved standard process document and template to aid in configuration management:
- (a) GRC-SW-7150.9, “Software Configuration Management”
 - (b) GRC-SW-TPLT-SCMP, “Software Configuration Management Plan Template”

4.3.2 Acquisition

- a. The purpose of the acquisition supporting activities is to conduct supplier management, to complete activities for product acceptance, and to complete activities for transitioning the acquired product into the project use.
 - b. The project **shall** fulfill the acquisition monitoring requirements as specified in the RMM for the software class in appendix C.3.
- (1) Appendix E highlights the following approved standard process documents to aid in acquisition monitoring:
- (a) GRC-SW-7150.4, “Software Project Monitoring and Control”
 - (b) GRC-SW-7150.15 “Software Acquisition Planning”

4.3.3 Off-The-Shelf

- a. The purpose of the Off-The-Shelf (OTS) supporting activities is to ensure successful acquisition of COTS, GOTS, or MOTS products. The OTS includes considerations of requirements satisfaction and sufficient documentation.
- (1) The project technical requirements for the OTS software **shall** be documented.

- (2) The OTS software **shall** be validated to the same level of confidence as developed software.
- (3) Follow-on support and maintenance of the OTS software **shall** be planned.
- (4) Licensing and intellectual property rights for the OTS software **shall** be ensured.
- (5) The OTS software **shall** be documented to the same level required of developed software.
- b. The project **shall** fulfill the OTS acquisition requirements as specified in the RMM for the software class in appendix C.3.

4.3.4 Risk Management

- a. The purpose of the risk management supporting activities is to identify potential problems and to manage their mitigation.
- b. The project **shall** fulfill the risk management activities of appendix C.8.
- (1) Refer to NPR 8000.4, “Agency Risk Management Procedural Requirements” for approved standard procedures.

4.3.5 Verification and Validation

- a. The purpose of the verification and validation supporting activities is to ensure fulfillment of requirements in the product, and to ensure the right product is built.
- b. The project **shall** fulfill the validation requirements as specified in the RMM for the software class in appendix C.9.
- (1) Refer to GLPR 7123.22, “Product Verification and/or Validation” for an approved standard procedure document.
- c. The project **shall** fulfill the verification requirements as specified in the RMM for the software class in appendix C.8.
- (1) Refer to GLPR 7123.22, “Product Verification and/or Validation” for an approved standard procedure document.
- d. Appendix E highlights the following approved standard process document and template to aid in inspection and peer reviews:
 - (1) Refer to GRC-SW-7150.12, “Formal Inspection and Peer Reviews”

4.3.6 Project Monitoring and Control

- a. The purpose of project monitoring and control (PMC) is to provide an understanding of the project’s progress so appropriate corrective action can be taken to ensure the integrity of work products. The PMC is to be performed periodically throughout the software project life-cycle after initial baselining of the SMP.
- b. The project **shall** fulfill the PMC requirements as specified in the RMM for the software class in appendix C.11.
- (1) Appendix E highlights the following approved standard process document to aid in PMC:
 - (a) GRC-SW-7150.4 “Software Project Monitoring and Control”

4.3.7 Metrics

- a. The purpose of the metrics supporting activity is to develop and utilize a measurement capability to support management information needs. Measurement activities at the software project level will assist in managing the project, assuring quality, and improving software engineering practices.
 - b. The project **shall** define metrics to address the four areas of software progress, software quality, software functionality, and requirements volatility.
 - c. The project **shall** record its characteristics and submit them into the software inventory.
 - d. The project **shall** fulfill the metrics requirements as specified in the RMM for the software class in appendix C.12.
- (1) Appendix E highlights the following approved standard process documents and templates to aid in the overall metrics activities:
- (a) GRC-SW-7150.7, “Managing the Measurement of Software Process and Product”
 - (b) GRC-SW-7150.8, “Performing the Measurement of Software Process and Product”
 - (c) GRC-SW-TPLT-MMMS, “Master Monthly Metrics Spreadsheet”

Chapter 5. Process Improvement

5.1 Determine Changes

a. The Project Manager, Project Chief Engineer, Lead Systems Engineer, and Software Project Lead (SPL) **shall** analyze the defined process documents and work products resulting from this procedure to determine if any changes or improvements to this procedure will make future documents easier to produce, more reliable, and/or of higher quality.

5.2 Submit Feedback

a. The SPL **shall** provide feedback from the analysis in section 6.1 to the SEPG if the analysis indicates changes and/or improvements in this procedure are warranted. This can be done by using the Feedback Form (located at Software@Glenn) or by meeting with the SEPG Lead.

b. Programs/projects **shall** provide comments/feedback regarding the GLPR to the CEO by use of the CPARS in accordance with GLPR 1270.1 for future updates.

Appendix A. Definitions

A.1 Acquisition. Contracting out the development of software or software components that will be incorporated into a deliverable product or system.

A.2 Code. See Software.

A.3 Commercial Off-The-Shelf (COTS) Software. Operating systems, libraries, applications, and other software purchased from a commercial vendor. Not customized for a particular project. Access to source code and documentation are often limited.

A.4 Configuration Management. A discipline applying technical and administrative direction, and control and surveillance to: Identify and document the functional and physical characteristics of a configuration item, control changes to those characteristics, record and report change processing and implementation status, and verify compliance with specified requirements.

A.5 Customer. The person or organization which defines the project objectives, key milestones, deliverables, and schedules.

A.6 Development. The process of translating a design into software components (code). The creation and debugging of a software product from design documentation.

A.7 Firmware. Computer programs and data loaded in a class of memory that cannot be dynamically modified during processing. The combination of hardware device, computer instructions and data that reside as read-only software on that device.

A.8 Freeware. Software that is available free of charge for personal use.

A.9 Government Off-The-Shelf (GOTS) Software. This refers to Government-created software, usually from another project. The software was not created by the current developers (see software reuse). Usually, source code is included and all documentation, including test and analysis results, is available. That is, the Government is responsible for the GOTS software to be incorporated into another system (definition from source document NASA-GB-8719.13).

A.10 Independent Verification and Validation (IV&V). A process whereby the products of the software development life-cycle phases are independently reviewed, verified, and validated by an organization that represents an acquirer of the software and is independent of the provider.

A.11 Legacy/Heritage. Software products (architecture, code, requirements) written specifically for one project and then, without prior planning during its initial development, found to be useful on other projects. See Software reuse.

A.12 Life-cycle. The period of time that starts when a software product is conceived and ends when the software is no longer available for use. The software life-cycle defined in this procedure includes

the following phases: concept, requirements, design, code and test, release and maintenance. The life-cycle of a product may be tailored to fit the resources associated with the project.

A.13 Modified Off-The-Shelf (MOTS) Software. When COTS, legacy, reuse, or heritage software is changed to a certain degree, usually more than 10 percent, then it is considered "modified." The changes can include all or part of the software products and may involve additions, deletions, and specific alterations. An argument can be made that any alterations to the code and/or design of an off-the-shelf software component constitutes "modification;" however, the common usage allows for some percentage of change before the off-the-shelf software is declared to be MOTS software. This may include the changes to the application shell and/or glueware to add or protect against certain features and not to the OTS software system code directly. See OTS software.

A.14 Off-The-Shelf (OTS) Software. Software not developed in-house or by a contractor for the specific project now underway. The software is general purpose or developed for a different purpose from the current project. The OTS software can include COTS, GOTS, MOTS, shareware, freeware, open source, and reuse, legacy, and/or heritage software.

A.15 Open Source. OTS usually developed outside the project team, sometimes by an indefinite community of independent collaborators, for which the source code may be available to the general public. Users of open source software are (generally) able to view the source code, alter, and redistribute open source software.

A.16 P (Center). Partial Center (i.e., per approved GRC defined processes that tailors the requirement(s) while still meeting a nonempty subset of the full requirement).

A.17 Requirements Traceability Matrix. A tool used to document traceability from requirements through design and testing.

A.18 Shareware. Software developed for the public domain, which can be used, or copied without infringing copyright.

A.19 Software. Computer programs, source code, source code listings, design details, algorithms, processes, flow charts, firmware, formulae, and related material that would enable the software or a functionally equivalent software to be reproduced, recreated, or recompiled, regardless of the form or media on which such information is recorded.

A.20 Software Assurance. The planned and systematic set of activities that ensure that software life-cycle processes and products conform to requirements, standards, and procedures. For NASA, this includes the disciplines of Software Quality (functions of Software Quality Engineering, Software Quality Assurance, and Software Quality Control), Software Safety, Software Reliability, Software Verification and Validation, and IV&V.

A.21 Software Classification. A designation given to software based on project characteristics as defined in NPR 7150.2. Software classes are labeled A, B, C, D, E, F, G, and H. The classification defines the requirements the project must meet during development of the software.

A.22 Software Development Folder (SDF). A software development folder could be used to collect the documentation and records associated with a development effort. The folder, which can be

electronic and/or paper, provides a centralized location for the developers to collect and maintain project-specific information. A development folder may contain the documentation and records described in Section 1.2 of this document, as well as development-related information such as e-mails, vendor information, procurement information, etc.

A.23 Software Deliverable(s). The agreed-upon software products to be provided to the customer. (See 'Software Products' below.)

A.24 Software Products. Software products consist of the documentation (plans, requirements, design, procedures, reports, etc.), code, executables, analysis results, intermediate analyses, and research, tests, models, simulators, prototypes, etc. created during the software development process. Many of these are for internal use in understanding and communicating what is to be created and how to go about the development process depending on the software classification.

A.25 Software Reuse. A software product developed for one use but having other uses or one developed specifically to be useable on multiple projects or in multiple roles on one project.

A.26 Validation. Validation provides objective evidence that the product meets the intended use of the product. Validation follows successful verification activities and may include system readiness reviews, test readiness reviews, operational readiness reviews, inspections, and testing.

A.27 Verification. Verification provides evidence that the design, or system, meets the input requirements. This evidence may consist of alternative calculations showing similarity with a proven design, peer or outside design reviews, analytical simulations, and test results.

Appendix B. Acronyms

BMS	Business Management System
CCB	Configuration Control Board
CEO	Chief Engineer's Office
CIO	Chief Information Officer
CM	Configuration Management
CMMI	Capability Maturity Model Integration
COTS	Commercial Off-The-Shelf
CSCI	Computer Software Configuration Item
CPARS	Corrective and Preventive Action Report System
DLE	Discipline Lead Engineer
GLPR	Glenn Procedural Requirements
GLWI	Glenn Work Instruction
GOTS	Government Off-The-Shelf
GRC	Glenn Research Center
HWCI	Hardware Configuration Item
KDP	Key Decision Point
LSE	Lead Systems Engineer
MOTS	Modified Off-The-Shelf
NPR	NASA Procedural Requirements
OTS	Off-The-Shelf
PCE	Project Chief Engineer
PM	Project Manager
PPAD	Program and Project Assurance Division
RMM	Requirements Mapping Matrix
RTM	Requirements Traceability Matrix
SAP	Software Acquisition Plan
SDF	Software Development Folder
SEPG	Software Engineering Process Group
SMAD	Safety and Mission Assurance Directorate
SMP	Software Management Plan
SPL	Software Project Lead
SRS	Software Requirements Specification
SWE	Software Engineering
TA	Technical Authority
WBS	Work Breakdown Structure

Appendix C. Requirements Mapping Matrix

- a. The Requirements Mapping Matrix (RMM) provides a mapping of all the requirements that projects need to comply with based upon the software's class and safety criticality. The purpose is to provide the project with a quick overview of all the requirements that it must meet. The project still needs to refer back to the NPR 7150.2 for the definitions and full text of the requirement (including applicable notes).
- b. The RMM presented here contains the same information as the RMM from appendix D of NPR 7150.2, but just groups the requirements by process areas to make it easier for the project to implement. It also excludes the requirements listed in NPR 7150.2 whose responsibilities are not at the project level.
- c. Classes A and B are omitted from this table in order to save space, but projects developing Class A or B software must comply with all of the requirements.
- d. Projects developing Class C, D, or E software need to comply with the requirements marked by an "X" or "P (Center)." Be sure to refer to the correct column in the RMM, based upon whether the software is safety critical or not.
- e. Projects developing Class F, G, or H software need to comply with the requirements marked by an "X" or "P (Center)."
- f. Projects need to implement a non-null subset of the full requirements for the items listed as "P (Center)." Projects should consult with their Software Technical Authority (TA) for these items and come to a consensus on what is appropriate for the project. The Software TA has final authority on the P (Center) requirements.

C.1 Management

Section of NPR	Requirement Descriptor**	SWE #	Responsibility	Safety Critical (Note 2)			NOT Safety Critical			Class F	Class G	Class H
				Class C	Class D	Class E	Class C	Class D	Class E			
Project Formulation	CMMI levels for class A, B, and C software	32	Project	P (Center)			P (Center)					
Project Formulation	Acquisition assessment	33	Project	X	X	X	X	X		X	X	
Project Formulation	Acceptance criteria	34	Project	X	X	X	X	X		X	X	
Project Formulation	Software processes	36	Project	X	X	X	X	P (Center)		X (not OTS)	P (Center)	
Compliance	Compliance matrix	125	Project	X	X	X	X	X	X	X	X	X
Compliance	Shall statements in this NPR	139	Project, Center	X	X	X	X	X	X	X	X	X
Compliance	Meeting "P (Center)"	140	Project, Center	X	X	X	X	X	X	X	X	X

C.2 Planning

Section of NPR	Requirement Descriptor**	SWE #	Responsibility	Safety Critical (Note 2)			NOT Safety Critical			Class F	Class G	Class H
				Class C	Class D	Class E	Class C	Class D	Class E			
SW Life-Cycle Planning	Software plans	13	Project	X	X	X	X	P (Center)	P (Center)	X	P (Center)	
SW Life-Cycle Planning	Execute planning	14	Project	X	X	X	X	X	X	X	P (Center)	

Section of NPR	Requirement Descriptor**	SWE #	Responsibility	Safety Critical (Note 2)			NOT Safety Critical			Class F	Class G	Class H
				Class C	Class D	Class E	Class C	Class D	Class E			
SW Life-Cycle Planning	Cost estimation	15	Project	X	X	X	X	P (Center)	P (Center)	X	P (Center)	
SW Life-Cycle Planning	Schedule	16	Project	X	X	X	X	P (Center)		X	P (Center)	
SW Life-Cycle Planning	Training	17	Project	X	X	X	X			X	P (Center)	
SW Life-Cycle Planning	Software development life-cycle or model	19	Project	X	X	X	X	P (Center)		X (not OTS)	P (Center)	
SW Life-Cycle Planning	Software classification	20	Project	X	X	X	X	X	X	X	X	X
SW Life-Cycle Planning	Software classification changes	21	Project	X	X	X	X	X	X	X	X	X
SW Life-Cycle Planning	Software assurance	22	Project	X	X	X	X	X		X	X	
SW Life-Cycle Planning	Software safety	23	Project	X	X	X	X	X	X	X	X	X
Verification and Validation	Verification planning	28	Project	X	X	X	X	P (Center)		X	P (Center)	
Verification and Validation	Validation planning	29	Project	X	X	X	X	P (Center)		X	P (Center)	
Software Operations, Maintenance, and Retirement	Document maintenance plan	74	Project	X	X	X	X			X	P (Center)	
Software Operations, Maintenance, and Retirement	Plan operations, maintenance and retirement	75	Project	X	X	X	X			X	P (Center)	

Section of NPR	Requirement Descriptor**	SWE #	Responsibility	Safety Critical (Note 2)			NOT Safety Critical			Class F	Class G	Class H
				Class C	Class D	Class E	Class C	Class D	Class E			
Software Configuration Management	Develop configuration management plan	79	Project	X	X	X	X	X		X	P (Center)	
Software Measurement	Objectives	90	Project	X	SO	SO	X			X (not OTS)	P (Center)	
Software Measurement	Software measurement areas	91	Project	P (Center) + SO	P (Center) + SO	P (Center) + SO	P (Center)			X (not OTS)	P (Center)	
Software Documentation Requirements	Software development /management plan	102	Project	P (Center) + SO	P (Center) + SO	P (Center) + SO	P (Center)	P (Center)		X (not OTS)	P (Center)	
Software Documentation Requirements	Software configuration management plan	103	Project	P (Center) + SO	P (Center) + SO	P (Center) + SO	P (Center)	P (Center)		X	P (Center)	
Software Documentation Requirements	Software test plan	104	Project	P (Center) + SO	P (Center) + SO	P (Center) + SO	P (Center)	P (Center)		X	P (Center)	
Software Documentation Requirements	Software maintenance plan	105	Project	SO	SO	SO				X	P (Center)	
Software Documentation Requirements	Software assurance plan	106	Project	X	X	X	X			X	X	
SW Life-Cycle Planning	Software safety plan	130	Project	X	X	X				X (if Safety Critical)	X (if Safety Critical)	X (if Safety Critical)
SW Life-Cycle Planning	IV&V Plan	131	IV&V Program	X (if selected for IV&V)	X (if selected for IV&V)	X (if selected for IV&V)	X (if selected for IV&V)					

Section of NPR	Requirement Descriptor**	SWE #	Responsibility	Safety Critical (Note 2)			NOT Safety Critical			Class F	Class G	Class H
				Class C	Class D	Class E	Class C	Class D	Class E			
SW Life-Cycle Planning	Independent software classification assessment	132	SMA organization	X	X	X	X	X	X	X	X	X
SW Life-Cycle Planning	Software safety determination	133	Project and SMA organization	X	X	X	X	X	X	X	X	X
SW Life-Cycle Planning	Safety-critical software requirements	134	Project	(Note 7)	(Note 7)	(Note 7)						
Software Peer Reviews/ Inspections	Peer review/ inspections of software plans	137	Project	P (Center) + SO	P (Center) + SO	P (Center) + SO	P (Center)			X (not OTS)	P (Center)	
Software Documentation Requirements	Software safety plan contents	138	Project	X	X	X				X (if Safety Critical)	X (if Safety Critical)	X (if Safety Critical)

C.3 Acquisition

Section of NPR	Requirement Descriptor**	SWE #	Responsibility	Safety Critical (Note 2)			NOT Safety Critical			Class F	Class G	Class H
				Class C	Class D	Class E	Class C	Class D	Class E			
Project Formulation	Supplier selection	35	Project	X	SO	SO	X			X	X	P (Center)
Project Formulation	Software milestones	37	Project	X	X	X	X	P (Center)		X (not OTS)	P (Center)	
Project Formulation	Acquisition planning	38	Project	X	X	X	X	X		X (not OTS)	P (Center)	
Off-The-Shelf (OTS) SW	COTS, GOTS, MOTS, etc.	27	Project	X	X	X	X			X	P (Center)	

Section of NPR	Requirement Descriptor**	SWE #	Responsibility	Safety Critical (Note 2)			NOT Safety Critical			Class F	Class G	Class H
				Class C	Class D	Class E	Class C	Class D	Class E			
Government Insight	Insight into software activities	39	Project	P (Center) + SO	P (Center) + SO	P (Center) + SO	P (Center)	P (Center)		X	P (Center)	
Government Insight	Access to software products	40	Project	P (Center) + SO	P (Center) + SO	P (Center) + SO	P (Center)			X	P (Center)	
Government Insight	Open source notification	41	Project	X	SO	SO	X			X	P (Center)	
Government Insight	Electronic access to Source code	42	Project	X	SO	SO	X			X	P (Center)	
Supplier Monitoring	Track change request	43	Project	P (Center) + SO	P (Center) + SO	P (Center) + SO	P (Center)			X	P (Center)	
Supplier Monitoring	Software measurement data	44	Project	X	X	X	X	P (Center)		X	P (Center)	
Supplier Monitoring	Joint audits	45	Project	X	SO	SO	X			X	P (Center)	
Supplier Monitoring	Software schedule	46	Project	X	X	X	X	X		X	P (Center)	
Supplier Monitoring	Traceability data	47	Project	X	SO	SO	X			X	P (Center)	
Supplier Monitoring	Solicitation	48	Project	X	X	X	X	P (Center)		X	P (Center)	

C.4 Requirements

Section of NPR	Requirement Descriptor**	SWE #	Responsibility	Safety Critical (Note 2)			NOT Safety Critical			Class F	Class G	Class H
				Class C	Class D	Class E	Class C	Class D	Class E			
Software Requirements Development	Document	49	Project	X	X	X	X	X	P (Center)	X	X	

Section of NPR	Requirement Descriptor**	SWE #	Responsibility	Safety Critical (Note 2)			NOT Safety Critical			Class F	Class G	Class H
				Class C	Class D	Class E	Class C	Class D	Class E			
Software Requirements Development	Software requirements	50	Project	X	X	X	X	X		X	X	
Software Requirements Development	Flow-down and derived requirements	51	Project	X	X	X	X			X (not OTS)	P (Center)	
Software Requirements Development	Bidirectional traceability	52	Project	P (Center) + SO	P (Center) + SO	P (Center) + SO	P (Center)			X (not OTS)	P (Center)	
Software Requirements Management	Manage requirements change	53	Project	X	X	X	X	X	P (Center)	X	X	
Software Requirements Management	Corrective action	54	Project	X	X	X				X	X	
Software Requirements Management	Requirements validation	55	Project	X	X	X	X			X	X	
Software Documentation Requirements	Software requirements specification	109	Project	P (Center) + SO	P (Center) + SO	P (Center) + SO	P (Center)	P (Center)		X	X	

C.5 Design

Section of NPR	Requirement Descriptor**	SWE #	Responsibility	Safety Critical (Note 2)			NOT Safety Critical			Class F	Class G	Class H
				Class C	Class D	Class E	Class C	Class D	Class E			
Software Design	Document design	56	Project	X	X	X	X			X (not OTS)	X (not OTS)	

Section of NPR	Requirement Descriptor**	SWE #	Responsibility	Safety Critical (Note 2)			NOT Safety Critical			Class F	Class G	Class H
				Class C	Class D	Class E	Class C	Class D	Class E			
Software Design	Software architecture	57	Project	P (Center) + SO	P (Center) + SO	P (Center) + SO	P (Center)	P (Center)		X (not OTS)	X (not OTS)	
Software Design	Detailed design	58	Project	SO	SO	SO				X (not OTS)	X (not OTS)	
Software Design	Bidirectional traceability	59	Project	SO	SO	SO				X (not OTS)	X (not OTS)	
Software Documentation Requirements	Software data dictionary	110	Project	P (Center) + SO	P (Center) + SO	P (Center) + SO	P (Center)			X (not OTS)	X (not OTS)	
Software Documentation Requirements	Software design description	111	Project	P (Center) + SO	P (Center) + SO	P (Center) + SO	P (Center)	P (Center)		X (not OTS)	X (not OTS)	
Software Documentation Requirements	Interface design description	112	Project	P (Center) + SO	P (Center) + SO	P (Center) + SO	P (Center)			X (not OTS)	X (not OTS)	

C.6 Code and Test

Section of NPR	Requirement Descriptor**	SWE #	Responsibility	Safety Critical (Note 2)			NOT Safety Critical			Class F	Class G	Class H
				Class C	Class D	Class E	Class C	Class D	Class E			
Software Implementation	Design into code	60	Project	X	X	X	X	X		X (not OTS)	X (not OTS)	
Software Implementation	Coding standards	61	Project	X	X	X				X (not OTS)	X (not OTS)	
Software Implementation	Unit test	62	Project	X	X	X	X	P (Center)		X (not OTS)	X (not OTS)	
Software Implementation	Version description	63	Project	X	X	X	X	P (Center)		X (not OTS)	X (not OTS)	
Software Implementation	Bidirectional traceability	64	Project	SO	SO	SO				X (not OTS)	X (not OTS)	
Software Implementation	Static analysis	135	Project	X	SO	SO	X					

Section of NPR	Requirement Descriptor**	SWE #	Responsibility	Safety Critical (Note 2)			NOT Safety Critical			Class F	Class G	Class H
				Class C	Class D	Class E	Class C	Class D	Class E			
Software Implementation	Validation of software development tools	136	Project	X (if Note 4 is true)	X (if Note 4 is true)	X (if Note 4 is true)	X (if Note 4 is true)	X (if Note 4 is true)				
Software Testing	Plan, procedures, reports	65	Project	X	X	X	X	P (Center)		X	P (Center)	
Software Testing	Perform testing	66	Project	X	X	X	X	X	P (Center) (Note 6)	X	P (Center)	
Software Testing	Verify implementation	67	Project	X	X	X	X			X	P (Center)	
Software Testing	Evaluate test results	68	Project	X	X	X	X	X		X	P (Center)	
Software Testing	Document defects and track	69	Project	X	X	X	X	P (Center)		X	P (Center)	
Software Testing	Models, simulations, tools	70	Project	X (if Note 5 is true)	X (if Note 5 is true)	X (if Note 5 is true)	X (if Note 5 is true)	X (if Note 5 is true)		X	P (Center)	
Software Testing	Update plans and procedures	71	Project	X	X	X	X			X	P (Center)	
Software Testing	Bidirectional traceability	72	Project	X	X	X	X	X		X	P (Center)	
Software Testing	Platform or hi-fidelity simulations	73	Project	X	X	X	X			X	P (Center)	
Software Documentation Requirements	Software test procedures	114	Project	X	X	X	X			X	X	
Software Documentation Requirements	Software test report	118	Project	P (Center) + SO	P (Center) + SO	P (Center) + SO	P (Center)			X	P (Center)	

C.7 Release and Maintenance

Section of NPR	Requirement Descriptor**	SWE #	Responsibility	Safety Critical (Note 2)			NOT Safety Critical			Class F	Class G	Class H
				Class C	Class D	Class E	Class C	Class D	Class E			
Software Operations, Maintenance, and Retirement	Implement plans	76	Project	X	X	X	X			X	P (Center)	
Software Operations, Maintenance, and Retirement	Deliver software products	77	Project	X	X	X	X	X	P (Center)	X	X	
Software Operations, Maintenance, and Retirement	As-built documentation	78	Project	X	X	X	X			X (not OTS)	X (not OTS)	
Software Documentation Requirements	Software users manual	115	Project	SO	SO	SO				X (not OTS)	X (not OTS)	
Software Documentation Requirements	Software version description	116	Project	P (Center) + SO	P (Center) + SO	P (Center) + SO	P (Center)	P (Center)		X (not OTS)	X (not OTS)	

C.8 Risk Management

Section of NPR	Requirement Descriptor**	SWE #	Responsibility	Safety Critical (Note 2)			NOT Safety Critical			Class F	Class G	Class H
				Class C	Class D	Class E	Class C	Class D	Class E			
Risk Management	Continuous risk management	86	Project	SO	SO	SO				X	P (Center)	

Section of NPR	Requirement Descriptor**	SWE #	Responsibility	Safety Critical (Note 2)			NOT Safety Critical			Class F	Class G	Class H
				Class C	Class D	Class E	Class C	Class D	Class E			
Software Peer Reviews/ Inspections	Requirements, test plans, design and code	87	Project	X	SO	SO	X			X (not OTS)	P (Center)	
Software Peer Reviews/ Inspections	Checklist, criteria and tracking	88	Project	P (Center) + SO	P (Center) + SO	P (Center) + SO	P (Center)			X (not OTS)	P (Center)	
Software Peer Reviews/ Inspections	Basic measurements	89	Project							X (not OTS)	P (Center)	
Software Documentation Requirements	Software inspection/peer review/ inspections	119	Project	P (Center) + SO	P (Center) + SO	P (Center) + SO	P (Center)			X (not OTS)	P (Center)	

C.9 Verification and Validation

Section of NPR	Requirement Descriptor**	SWE #	Responsibility	Safety Critical (Note 2)			NOT Safety Critical			Class F	Class G	Class H
				Class C	Class D	Class E	Class C	Class D	Class E			
Verification and Validation	Verification results	30	Project	X	X	X	X	X		X	P (Center)	
Verification and Validation	Validation results	31	Project	X	X	X	X	X		X	P (Center)	

C.10 Configuration Management

Section of NPR	Requirement Descriptor**	SWE #	Responsibility	Safety Critical (Note 2)			NOT Safety Critical			Class F	Class G	Class H
				Class C	Class D	Class E	Class C	Class D	Class E			
Software Configuration Management	Track and evaluate changes	80	Project	X	X	X	X	P (Center)		X	P (Center)	

Section of NPR	Requirement Descriptor**	SWE #	Responsibility	Safety Critical (Note 2)			NOT Safety Critical			Class F	Class G	Class H
				Class C	Class D	Class E	Class C	Class D	Class E			
Software Configuration Management	Identify software configuration items	81	Project	X	X	X	X	X		X	P (Center)	P (Center)
Software Configuration Management	Authorizing changes	82	Project	X	X	X				X	P (Center)	P (Center)
Software Configuration Management	Maintain records	83	Project	X	X	X	X			X	P (Center)	P (Center)
Software Configuration Management	Configuration audits	84	Project	SO	SO	SO				X	P (Center)	
Software Configuration Management	Implement procedures	85	Project	X	X	X	X	P (Center)	P (Center)	X	P (Center)	P (Center)

C.11 Project Monitoring and Control

Section of NPR	Requirement Descriptor**	SWE #	Responsibility	Safety Critical (Note 2)			NOT Safety Critical			Class F	Class G	Class H
				Class C	Class D	Class E	Class C	Class D	Class E			
SW Life-Cycle Planning	Reviews	18	Project	X	X	X	X	X		X	P (Center)	
SW Life-Cycle Planning	Plan tracking	24	Project	X	X	X	X	P (Center)		X	P (Center)	
SW Life-Cycle Planning	Corrective action	25	Project	X	SO	SO	X			X	P (Center)	
SW Life-Cycle Planning	Changes	26	Project	X	X	X	X			X	P (Center)	

Section of NPR	Requirement Descriptor**	SWE #	Responsibility	Safety Critical (Note 2)			NOT Safety Critical			Class F	Class G	Class H
				Class C	Class D	Class E	Class C	Class D	Class E			
Software Documentation Requirements	Software change request/problem report	113	Project	P (Center) + SO	P (Center) + SO	P (Center) + SO	P (Center)			X (not OTS)	X (not OTS)	

C.12 Metrics

Section of NPR	Requirement Descriptor**	SWE #	Responsibility	Safety Critical (Note 2)			NOT Safety Critical			Class F	Class G	Class H
				Class C	Class D	Class E	Class C	Class D	Class E			
Software Measurement	Collection and storage	92	Project	X	SO	SO	X			X (not OTS)	P (Center)	
Software Measurement	Analyze data	93	Project	P (Center) + SO	P (Center) + SO	P (Center) + SO	P (Center)			X (not OTS)	P (Center)	
Software Measurement	Report analysis	94	Project	P (Center) + SO	P (Center) + SO	P (Center) + SO	P (Center)			X (not OTS)	P (Center)	
Software Documentation Requirements	Software metrics report	117	Project	P (Center) + SO	P (Center) + SO	P (Center) + SO	P (Center)			X (not OTS)	X (not OTS)	

C.13 Legend and Notes

X - "Required" - Project is required to meet the requirement as written.

X (not OTS) - Project is required to meet, except for off-the-shelf software.

P (Center) - Per approved Center defined process which meets a nonempty subset of the full requirement.

SO - "Safety Only" - Project is required to meet this requirement to the extent necessary to satisfy safety critical aspects of the software.

P (Center) + SO - Project is required to meet this requirement to the extent necessary to satisfy safety critical aspects of the software and follow TA-defined P (Center) processes for other aspects of the software.

Blank Space - Project is not required to meet the requirement, but is encouraged to consider its usefulness to the project.

Note 1 - Reserved

Note 2 - There are additional safety requirements in NASA-STD-8719.13, NASA Software Safety Standard.

Note 3 - NASA Headquarters' Chief, Safety and Mission Assurance has co-approval on any waiver or deviation decided at the HQ level that involves safety-critical software. NASA Headquarters' Chief Medical Officer has co-approval on any waiver or deviation decided at the HQ level that involves software with health and medical implications. Waivers and deviations decided at the Center level are to follow similar protocol when software safety critical or health and medical issues are involved.

Note 4 - When software development tools are used to develop or maintain Class C, or safety-critical software.

Note 5 - When software models, simulations, and analysis tools are used to perform qualification of flight software or flight equipment.

Note 6 - No test plans are required but the project is to perform software testing.

Note 7 - This NPR does not require SWE-134 for Classes C thru E and Safety Critical; however, it is highly recommended that software within this category use SWE-134 as a checklist to support the identification of safety-related risks and their mitigations.

Appendix D. Standard Process Tailoring Guidelines

D.1 NASA selected the Capability Maturity Model Integration – Development (CMMI-DEV) process model for its software improvement activities under the Engineering Excellence Initiative. This process model supports the progressive development of software expertise, which in turn is appraised at various capability or maturity levels. The GRC's SEPG developed a series of standard process model documents (see appendix E) to assist all software developers in meeting these levels by combining specific and generic practices recommended by the model with GRC best practices. The SEPG acknowledges that each project is unique and might benefit by a tailoring of the standard process description into a defined process. The purpose of these guidelines is to assist a project in converting/tailoring a GRC standard process into a defined process that is specific to the project's unique makeup. GRC standard processes are available from the Software@Glenn Web site (<http://software.grc.nasa.gov>).

D.2 When following the tailoring guidelines below, document the rationale for any tailoring decisions being proposed to the standard process document(s). This documentation will help in clarifying why the change is being proposed, and it will help others understand the change. It will also be useful to future projects, and will provide the SEPG with possible process improvement ideas.

D.3 Review the process against the applicability matrix in NPR 7150.2 (appendix C - Requirements Mapping Matrix). Check the comments balloons that are in the right hand margins of the electronic versions of the standard process document(s). These comments help to identify which parts of the process trace back to the NPR. Requirements that are marked as required by NPR 7150.2 (denoted by the letter 'x' that appears in the box next to the requirement, in the appendix C) must remain in the process. Requirements that are not required by NPR 7150.2 (denoted by an empty box) should be analyzed to determine if the project would benefit by optionally meeting all or some of the requirement. Requirements that are marked as P (Center) should be analyzed to determine what the project needs to do to meet the nonempty subset of the requirement according to paragraph 4.1.4 of this document.

D.4 Review the process steps to determine if the order is appropriate for the project. Rearrange as necessary.

D.5 Review the process steps to determine if they are all necessary for the project. If a step is not necessary, mark it as N/A.

D.6 Review the inputs and outputs of each step. Enter project specific information where it is needed.

D.7 Obtain concurrence on the tailored process from your Software Technical Authority (TA).

Appendix E. Standard Guidance

E.1 The following standard process documents and templates are directly referenced by this procedure and can assist projects in complying with NPR 7150.2 requirements. These documents can be found on the Software@Glenn Web site (<http://software.grc.nasa.gov>).

E.2 Approved Standard Processes (See Software@Glenn for latest versions and additional resources):

GRC-SW-7150.3, “Software Project Planning”

This standard process assists in planning the software project; planning the monitoring of the software project; preparing the SMP; starting the Software Development Folder; determining software classification, criticality, and compliance; establishing required documentation and estimates; selecting a life-cycle; developing a schedule; planning software sources, training, inspections, reviews, metrics, risks; and baselining the SMP.

GRC-SW-7150.4, “Software Project Monitoring and Control”

This standard process assists in the collection of metrics, performing risk assessments, determining project progress, tracking project activities, involving stakeholders, tracking supplier activities, evaluating products, reviewing and reporting status, performing risk mitigation, replanning, archiving, and generating lessons learned.

GRC-SW-7150.5, “Requirements Development”

This standard process assists in defining stakeholders, high-level requirements, the RTM, logical decomposition, documenting, and baselining requirements.

GRC-SW-7150.6, “Requirements Management”

This standard process assists in identifying new or modified requirements, impact assessments, change processes, change implementation, and tracking.

GRC-SW-7150.7, “Managing the Measurement of Software Process and Product”

This standard process assists in defining metrics goals and indicators and establishing the collection and reporting procedures.

GRC-SW-7150.8, “Performing the Measurement of Software Process and Product”

This standard process assists in gathering, analyzing, reporting and storing metrics.

GRC-SW-7150.9, “Software Configuration Management”

This standard process assists in planning SCMP activities. It includes help with configuration identification, configuration baselines, configuration control, status accounting, audits and reviews.

GRC-SW-7150.10, “Transition of Software to a Higher Classification”

This standard process assists in evaluating proposed reuse of software at a higher classification level, including performance evaluation, risk assessments, documentation collection, and testing.

GRC-SW-7150.11, “Preparing Software for Release”

This standard process assists in ensuring that a software product is ready for release prior to performing GLPR 2210.1, “Software Release.”

GRC-SW-7150.12, “Formal Inspection and Peer Reviews”

This standard process assists in the planning, execution, and completion of a formal inspection. The process also provides guidelines on how to modify the formal inspection process for a peer review.

GRC-SW-7150.13, “Process and Product Quality Assurance”

This standard process assists in defining and executing the software assurance plans, estimates, activities, team formation, assessments, reviews, and audits.

GRC-SW-7150.14, “Software Acquisition Statement of Work Guidelines”

This standard process assists in the development of a statement of work to ensure that a project addresses NPR 7150.2 acquisition requirements and the determination of the required documentation to be developed by a contractor.

GRC-SW-7150.15, “Software Acquisition Planning”

This standard process assists in planning for software sources, supplier agreements, contract types, project-supplier interactions, acceptance criteria, and product transfer planning.

GRC-SW-7150.16, “Software Estimating”

This standard process assists in estimating the cost and schedule of a software project.

E.3 Approved Standard Templates:

GRC-SW-TPLT-IDD, “Interface Design Description Template”

This standard template describes the interface characteristics of one or more systems, subsystems, Hardware Configuration Items (HWCIs), Computer Software Configuration Items (CSCIs), manual operations, or other system components, and is used when the developer is tasked to define and record the interface design of one or more systems or subsystems.

GRC-SW-TPLT-RTM, “Requirements Traceability Matrix Template”

This standard template assists the project in developing a traceability matrix. It includes help with generating a matrix to trace requirements back to the project objectives identified in the Software Management Plan and forward through the remainder of the project life-cycle phases.

GRC-SW-TPLT-SAP, “Software Assurance Plan Template”

This standard template describes Software Assurance tasks and resources required to assure that software being developed meets the technical requirements of the project, operates safely, and is developed according to applicable standards and procedural requirements.

GRC-SW-TPLT-SCMP, “Software Configuration Management Plan (SCMP) Template”

This standard template assists the project in developing a SCMP by describing the format and content of the SCMP.

GRC-SW-TPLT-SDD, “Software Design Description Template”

This standard template describes the design of a Computer Software Configuration Item (CSCI). It describes the CSCI-wide design decisions, the CSCI architectural design, and the detailed design needed to implement the software and is used when the developer is tasked to define and record the design of a CSCI.

GRC-SW-TPLT-SMP, “Software Management Plan (SMP) Template”

This standard template assists the project in developing a SMP by describing the format and content of the SMP. It includes help with project organization, managerial processes, technical processes, and supporting processes.

GRC-SW-TPLT-SMntP, “Software Maintenance Plan Template”

This standard template describes the activities planned for releasing, maintaining, upgrading, migrating, and retiring a Computer Software Configuration Item (CSCI).

GRC-SW-TPLT-SRS, “Software Requirements Specification (SRS) Template”

This standard template assists the project in developing a SRS by describing the format and content of the SRS. It includes help with functional requirements, interfaces, data, safety, security, privacy, environment, computer resources, design constraints, and related requirements.

GRC-SW-TPLT-SSP, “Software Safety Plan (SSP) Template”

This standard template assists the project in developing a SSP by describing the format and content of the SSP. It documents project’s software safety scopes and objectives, software safety management structure and tasks, and software safety analyses for the development of safety-critical software.

GRC-SW-TPLT-STP, “Software Test Plan Template”

This standard template describes plans for testing of Computer Software Configuration Items (CSCIs) and software systems. It describes the software test environment to be used for the testing, identifies the tests to be performed, and provides schedules for test activities.

GRC-SW-TPLT-STPr, “Software Test Procedure Template”

This standard template describes the test preparations, test cases, and test procedures to be used to perform testing of a Computer Software Configuration Item (CSCI) or a software system or subsystem.

GRC-SW-TPLT-STR, “Software Test Report Template”

This standard template is a record of the qualification testing performed on a Computer Software Configuration Item (CSCI), a software system or subsystem, or other software-related item.

GRC-SW-TPLT-SUM, “Software Users Manual Template”

This standard template tells a hands-on software user how to install and use a Computer Software Configuration Item (CSCI), a group of related CSCIs, or a software system or subsystem. It may also cover a particular aspect of software operation, such as instructions for a particular position or task.

GRC-SW-TPLT-SVD, “Software Version Description Template”

This standard template contains the format and content preparation instructions for the SVD and is used when the developer is tasked to identify and record the exact version of software to be delivered to a user, support, or other site.

GRC-SW-TPLT-MMMS, “Master Monthly Metrics Spreadsheet”

This standard template assists the software team personnel in collecting and reporting monthly metrics information to higher level management. It is an excel spreadsheet for collecting project metrics that are aligned with Center- and Agency-level metrics goals.